Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергетиці

Лабораторна робота № 3

з курсу: «*Розробка програмного*

*забезпечення мобільних пристроїв*»

**Виконав:**  
студент 4-го курсу,  
групи ТВ-13  
Романін Анатолій Олександрович

Посилання на GitHub репозиторій:https://github.com/JIAIM/mobile\_software

**Перевірив:**

Недашківський О.Л.

Київ 2024/2025

Лабораторна робота № 3

**Теоретичні відомості:**

**Розрахунок прибутку від сонячних електростанцій** базується на аналізі генерації сонячної енергії, з урахуванням встановленої системи прогнозування потужності, яка дозволяє підвищити точність передбачення майбутнього енергетичного виробництва. Сонячна енергія є одним із найперспективніших альтернативних джерел, і технології прогнозування є ключовими для максимізації прибутковості цього виду електростанцій.

#### Основні Параметри для Розрахунку Прибутку

1. **Інсоляція (сонячне випромінювання)**: залежить від кліматичних умов, часу доби, сезону, а також від географічного розташування електростанції.
2. **Потужність установок**: визначає кількість енергії, яку здатна генерувати електростанція. Встановлена потужність зазвичай вимірюється в кіловатах (кВт).
3. **ККД панелей**: коефіцієнт корисної дії (ККД) впливає на загальну генерацію енергії.
4. **Вартість електроенергії**: визначає прибуток від продажу виробленої енергії. Залежить від тарифів ринку електроенергії.

#### Методи Прогнозування Потужності

Система прогнозування сонячної потужності дозволяє аналізувати історичні дані, такі як погодні умови та рівень інсоляції, що підвищує точність розрахунків. За допомогою математичних моделей і машинного навчання можна побудувати прогнози потужності на певний період, що дозволяє управляти генерацією та фінансовими потоками більш ефективно.

**Контрольний приклад** описує базовий розрахунок для користувача, де застосовуються основні вхідні параметри для визначення прогнозованого прибутку.

**Завдання :**

Створіть мобільний калькулятор розрахунку прибутку від сонячних електростанцій з встановленою системою прогнозування сонячної потужності (див. приклад Задача 1).

**Хід виконання:**

Для обчислення формул, необхідно було знайти інтеграл, його ми знаходимо в наступних функціях. Також для більш зручного виводу створена функція round, що округлює значення до сотих.

private fun calculateIntegral(a: Double, b: Double, Pc: Double, sigma: Double): Double {  
 val step = 0.001  
 var sum = 0.0  
 var x = a  
  
 while (x <= b) {  
 sum += normalDistribution(x, Pc, sigma) \* step  
 x += step  
 }  
  
 return sum  
 }  
  
 private fun normalDistribution(p: Double, Pc: Double, sigma: Double): Double {  
 return (1 / (sigma \* *sqrt*(2 \* *PI*))) \* *exp*(-0.5 \* ((p - Pc) / sigma).*pow*(2))  
 }  
  
 private fun round(x: Double): Double {  
 return BigDecimal(x).setScale(2, RoundingMode.*HALF\_UP*).toDouble()  
 }  
}

Далі у наступних функція прописана логіка обрахунку прибутку електростанції:

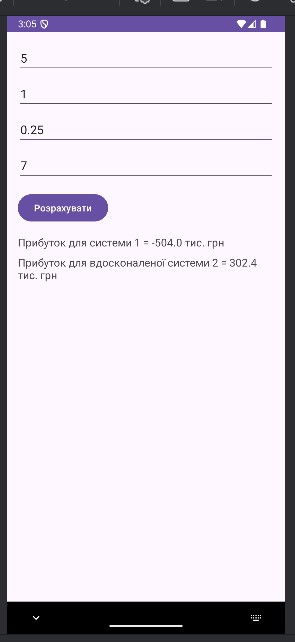
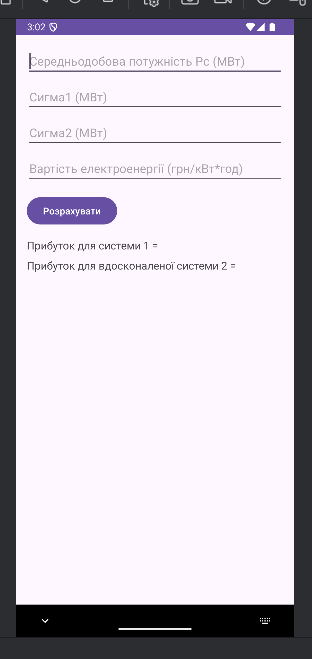
private fun calculatePribytok(Pct: Double, vartistt: Double, sigma: Double): Double {  
 val Pc = Pct  
 val vartist = vartistt  
 val sigmaW1 = findPercentEnergy(Pc, sigma)  
 val W1 = Pc \* 24 \* sigmaW1  
 val P1 = W1 \* vartist  
 val W2 = Pc \* 24 \* (1 - sigmaW1)  
 val shtraf = W2 \* vartist  
 val prib1 = P1 - shtraf  
  
 return round(prib1)  
}  
  
private fun findPercentEnergy(Pc: Double, sigma: Double): Double {  
 val a = 4.75 // нижня межа інтегрування  
 val b = 5.25 // верхня межа інтегрування  
 return round(calculateIntegral(a, b, Pc, sigma))  
}

Дані зчитуємо з вводу користувача, до кожного елемента звертаємся за його id

class MainActivity : AppCompatActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val inputPc = findViewById<EditText>(R.id.*inputPc*)  
 val inputSigma1 = findViewById<EditText>(R.id.*inputSigma1*)  
 val inputSigma2 = findViewById<EditText>(R.id.*inputSigma2*)  
 val inputVartist = findViewById<EditText>(R.id.*inputVartist*)  
 val calculateButton = findViewById<Button>(R.id.*calculateButton*)  
 val resultSigma1 = findViewById<TextView>(R.id.*resultSigma1*)  
 val resultSigma2 = findViewById<TextView>(R.id.*resultSigma2*)  
  
 calculateButton.setOnClickListener **{** val Pc = inputPc.*text*.toString().*toDoubleOrNull*() ?: 0.0  
 val sigma1 = inputSigma1.*text*.toString().*toDoubleOrNull*() ?: 0.0  
 val sigma2 = inputSigma2.*text*.toString().*toDoubleOrNull*() ?: 0.0  
 val vartist = inputVartist.*text*.toString().*toDoubleOrNull*() ?: 0.0  
  
 val prib1 = calculatePribytok(Pc, vartist, sigma1)  
 val prib2 = calculatePribytok(Pc, vartist, sigma2)  
  
 resultSigma1.*text* = "Прибуток для системи 1 = $prib1 тис. грн"  
 resultSigma2.*text* = "Прибуток для вдосконаленої системи 2 = $prib2 тис. грн"  
 **}** }

**Перевірка:**

Для перевірки результатів введемо початкові дані із контрольного прикладу:



Результати аналогічні

**Висновок**

У результаті виконання лабораторної роботи було розроблено мобільний калькулятор, що дозволяє розраховувати прибуток від сонячних електростанцій на основі прогнозованої потужності. Калькулятор використовує нормальний розподіл для розрахунку ймовірності отримання певного рівня потужності (Pc), враховуючи два показники (σ1 і σ2). Для кожної системи було визначено обсяги енергії, що можуть бути згенеровані та втрати, зокрема штрафи за недоотриману енергію. Кінцевий розрахунок прибутку допомагає оцінити фінансову ефективність кожної системи.

Під час виконання роботи було засвоєно основи програмування мовою Kotlin, зокрема, обробку введених даних, використання математичних функцій для інтеграції та округлення результатів для зручного виведення.